User Manual

FAGOR

FAGOR AUTOMATION
FEEDBACK DISPLAY PC BOARD

Fagor Automation S. Coop.
Barrio San Andrés, s/n.
20500 - Mondragón

Tel.: (34) 43 79 95 11
Fax: (34) 43 79 17 12
Internet: www.fagorautomation.mcc.es
E-mail: marketing-info@fagorautomation.mcc.es

The information contained in this manual may be subject to change due to technical modifications.

Fagor Automation S. Coop. reserves the right to modify the contents of this manual without prior notice.

Contents **Page**

Appendices **Page**

Figures **Page**

1. GENERAL INFORMATION

1.1 Introduction

The Fagor DRO board is used for connecting a compatible PC with up to four position feedback devices and a touch probe. It also has four digital input signals and a further four output signals for carrying out automation processes.

The software supplied will furthermore enable you to make full use of all your board's functions in your own particular application.

1.2 Computer requirements

- At least 640KB memory
- DOS version 3.0 or later
- 2 x 8 or 16 bit ISA slots free
- Sufficient current in the PC power supply (+5V ~ 1A, +12V ~ 1A, -12V ~ 0mA)
- C compiler

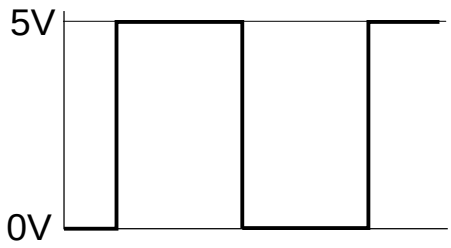
1.3 Specification of the boards

1.3.1 General characteristics

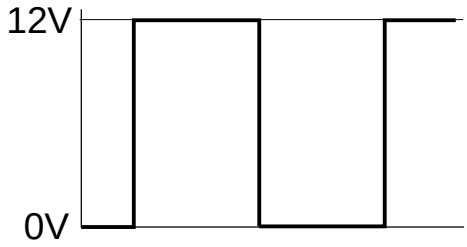
- Operating temperature : 0ºC ÷ +45º C
- Storage temperature : -25º ÷ 70º C
- Relative humidity : 20% ÷ 80%
- Weight : 250 g.
- Current consumption : +5V, 50mA typical, 1A maximum
  +12V, 50mA typical, 1A maximum
  -12V, 0mA maximum

1.3.2 DRO board

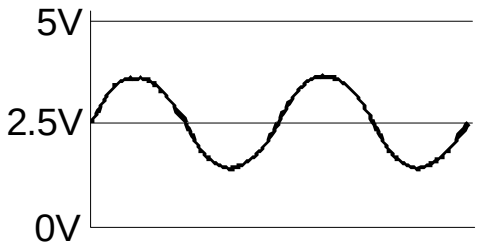
- Feedback inputs for four axes, with each axis consisting of:
    - A and B feedback signals, plus their complemented signals.
    - Home signal and its complemented signal.
- Internal 32-bit counter for each axis.
- Each axis can be configured for the following types of signals
    - Differential or non-differential TTL



    - Differential or non-differential HTL



    - Voltage modulated differential 1Vpp Sinewave



- Input frequency of signals A and B :

  - TTL and HTL signals 250KHz maximum
  - 1Vpp sine wave signals 50KHz maximum

- Power supply outputs for the feedback systems.

  - +5V (Protected against short-circuits) for TTL and 1Vpp sine wave signals
  - +12V (Protected against short-circuits) for HTL signals

- The particular setting between differential and non-differential signals is done externally, using the proper surface speed outputs for each type of signal for this purpose:

  - Surface speed output +1.5V for non-differential TTL[(1)]
  - Surface speed output +5V for non-differential HTL

- Probe input consisting of:

  - Probe contact input
  - +5V and 15mA output for activating its led

1.3.3 Input and output board

- Four optoinsulated inputs for general use. The operating voltages of the inputs will be between 0V and +24V (±25%) so that the separation threshold between 0 and 1 will be around +6V.

  - Minimum DC voltage 18V
  - Maximum DC voltage 30V

(1) See the signals of the counting board connector

- Four optoisolated outputs with solid state relay with normally open contact. The main features are as follows:

  - Maximum load intensity 225mA
  - Maximum leakage intensity 1µA
  - Maximum AC or DC voltage 40V
  - Minimum isolating voltage 1500V
  - Maximum resistance of the contact 5Ω
  - Maximum activation time 3ms
  - Maximum deactivation time 3ms

1.4 Block diagram

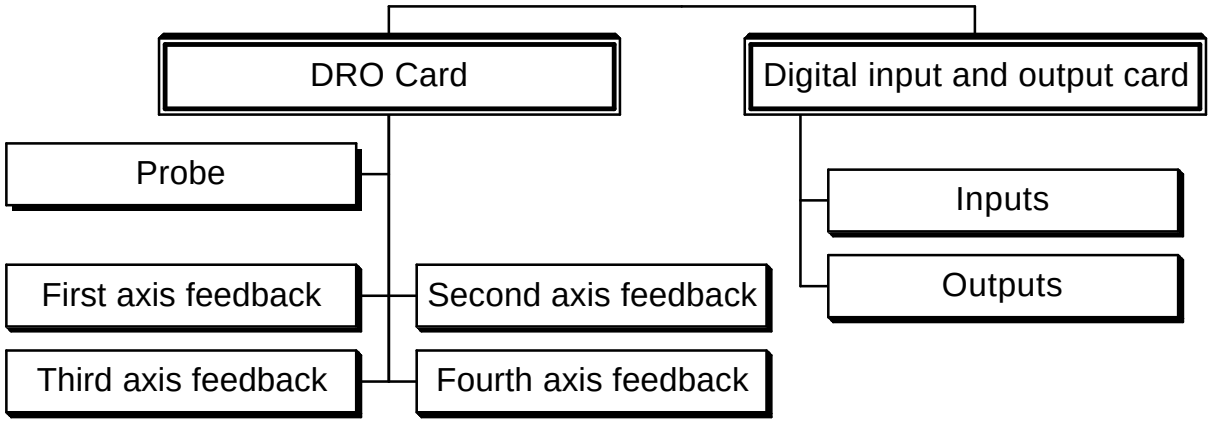


Figure 1, Block diagram

2. INSTALLATION

2.1 Preparation

You should have found the following items in the package:

- 1 DRO board
- 1 input and output board
- 2 diskettes with the software
- 1 37-pin male connector
- 1 15-pin male connector
- this manual

Check the quantity and condition of the components before going on, and if any of these is missing or in bad condition get in touch with your distributor or the closest Fagor Automation customer service department.

2.2 Description of the boards

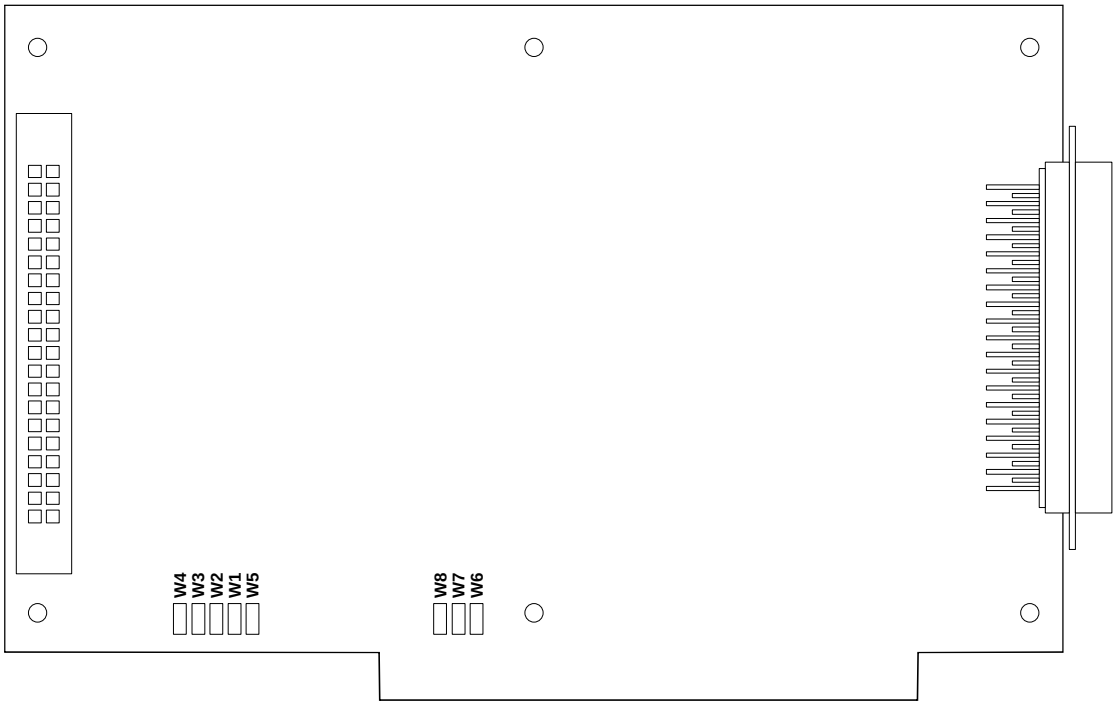


Figure 2, DRO board

Components of interest, numbered from left to right:

a) Expansion connector for the digital input and output connector
b) Jumpers for selecting the board address.
c) Jumpers for selecting the interruptions used by the board
d) External connector for connecting the feedback devices and the probe

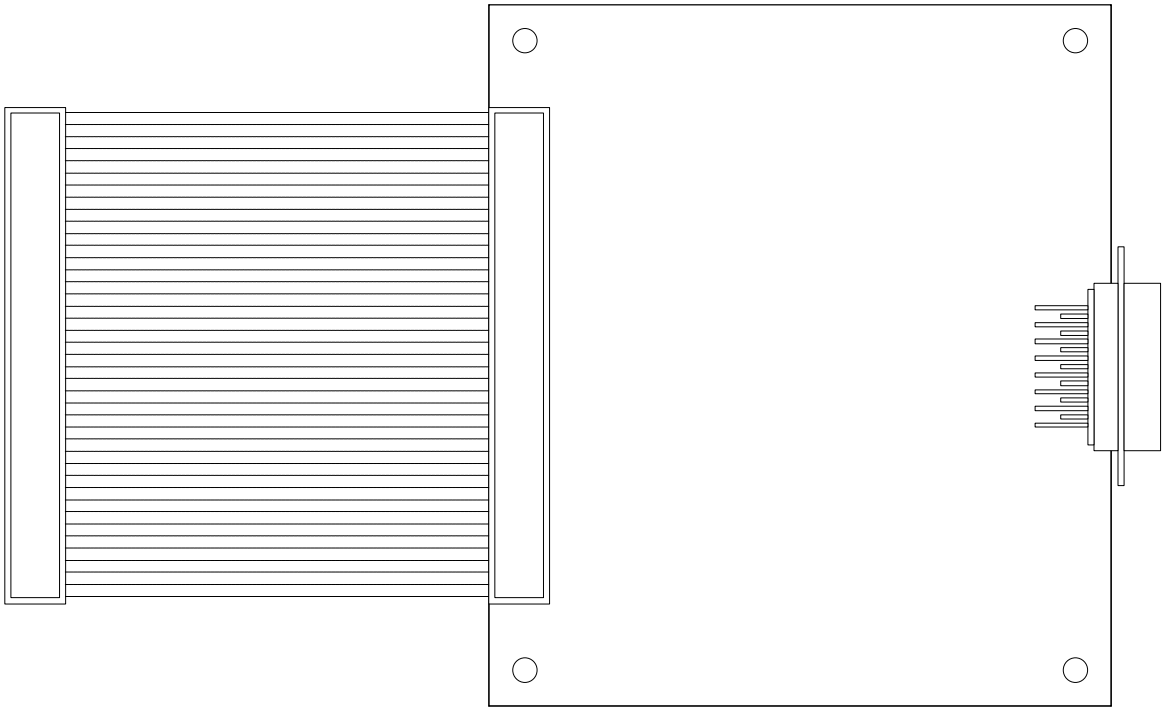


Figure 3, Digital input and output board

Components of interest, numbered from left to right

a) Connector for joining to the DRO board
b) External connector for connecting the digital inputs and outputs

Note.- This board uses an ISA slot, even though its connector is not used, since the signals are sent to it from the DRO board.

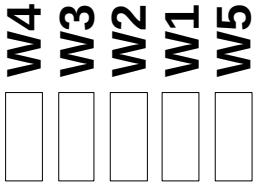
2.3 Setting the counting board

2.3.1 Selecting the access address

The program needs to know the address where the board is located, to be able to access the counters and other resources supplied.

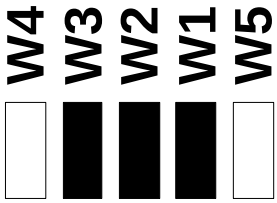
The board can be configured in any address within a range of from 100 to 3E0 hexadecimal. It is first necessary to know which addresses are not occupied by the system or by other boards. To do this, see 'APPENDIX A, PC input/output address map' and the configurations of the boards that you already have fitted in the computer.

For selecting the address, the board has five jumpers, W1 to W5, which when **not jumpered** (without the metal hood) mean the following :



W1	20 Hex
W2	40 Hex
W3	80 Hex
W4	100 Hex
W5	200 Hex

The default configuration is address **300 Hex** :

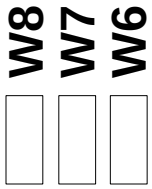


2.3.2 Selection of the interrupt number

To handle the references (markers) of the feedback devices and carry out treatment of the probe one has to work in real time and to do this, interrupts have to be used.

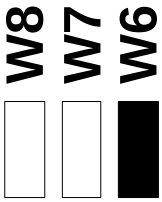
The board can work with one of three possible interrupt levels. You have to choose one that is not being used by the system or by another board, or even so, one which is not being used within the same application used by the DRO board. Bear in mind that if you are not in the feedback device reference search mode or operating with the probe, the interrupt is not being used by the program. For this purpose see 'APPENDIX B, System interrupts'.

To select the interrupt you are going to use there are three jumpers, W6 to W8: **jumpering** one of these means the following:



W6	IRQ 7
W7	IRQ 5
W8	IRQ 3

The default selection is interrupt **IRQ 7** :



2.4 Fitting the boards

WARNING!
Switch off the PC before fitting or removing the boards, and before connecting or disconnecting the feedback and input output cables.

1. Switch off the computer, and any peripherals connected to it (printers, monitors, etc.).

2. Remove the cover of the computer's central processing unit (See the computer users' manual if required).

3. Touch the computer ground screw or put on an anti-static bracelet connected to ground.

4. Locate the expansion slots, at the back of unit, and choose two consecutive free ISA slots. If there are none, make some space by changing the position of other boards.

5. Loosen the screws from the expansion slot covers and take out the metal plates (keep the screws, which will be needed later on).

6. Hold the DRO board by the top, taking care not to touch the components. Align the board retainer plate with the window of the expansion slot. Align the part of the board that sticks out (with gold lines) with the connector of the base. Press the board firmly onto the connector until it is securely fitted.

7. Do the same thing with the input and output board, bearing in mind that this is not connected to the base.

8. Fit and tighten the screws in the hole over the retainer plates.

9. Fit the cover on the unit. Connect the cables for feedback and digital signals (this can be done later). Lastly switch on the computer.

The installation of the boards is now complete, so go on to install the programs.

2.5 Copying the libraries and example programs

The programs included with the DRO board are:

- fdrodos.lib, library with the initialization functions and access to the boards for DOS applications.

- fdrow16.dll, dynamic library for 16-bit Windows applications.

- Demo programs for building an application with the board.

We recommend making a backup copy of the diskettes supplied, for example with the MS-DOS DISKCOPY utility (See MS-DOS users' manual for the operations needed).

To copy the programs only for DOS applications onto your hard disk, type, from the command line of MS-DOS :

<*diskette drive*>**:\INSTALL** <*target drive*>**:**

Where : <*diskette drive*> is the drive where the diskette is placed (A or B)
<*target drive*> is your hard disk drive (C to Z)

After completing the installation of files and programs, these will be found in directory ***FAGORDRO***.

To copy the programs required for generating DOS and Windows 3x applications, select 'File -> Execute' from the Program or File Manager in Windows, and type :

<*diskette drive*>**:\SETUP**

Then follow the instructions given in the program.

Within the installation directory, apart from the program FDEMOW16.EXE, as a demo in Windows, the directories DOS and Win16 are created, containing the files stated above.

For the demo application to operate, the following files and programs will also be copied if this is necessary and there is no later version of the same.

In the application directory:

FDROW16.INI	board configuration data
MESSAGES.TXT	text messages for the application software
MSG_ENG.TXT	text messages in English language
MSG_ESP.TXT	text messages in Spanish language
MSG_XXX.TXT	text messages in different languages

In the Windows\System directory:

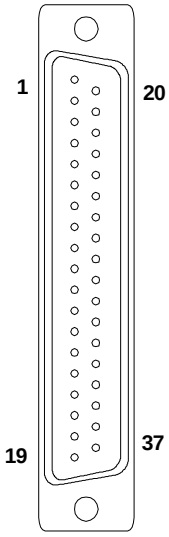
FDROW16.DLL	extension of the application for handling the board
VSHARE.386	
STKIT416.DLL	
VB40016.DLL	
OC25.DLL	

OC25ESP.DLL
OLE2.DLL
TYPELIB.DLL
OLE2DISP.DLL
OLE2PROX.DLL
OLE2CONV.DLL
STORAGE.DLL
COMPOBJ.DLL
OLE2.REG
OLE2NLS.DLL
STDOLE.TLB
SCP.DLL
VAEN21.OLB
CTL3DV2.DLL
VB4ES16.DLL
TABCTL16.OCX
THREED16.OCX
ANIBTN16.OCX
COMDLG16.OCX components of Windows and Visual Basic required for the demo program.

2.6 Description of the connectors

2.6.1 Feedback and probe connector

Female SUBD type connector with 37 pins, and the following signal allocation:

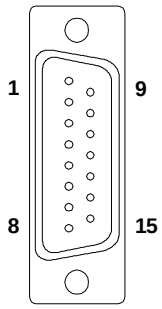


Pin	Signal	Pin	Signal
1	Signal A from axis 1	20	Signal /A from axis 1
2	Signal B from axis 1	21	Signal /B from axis 1
3	Home from axis 1	22	Home from axis 1
4	Signal A from axis 2	23	Signal /A from axis 2
5	Signal B from axis 2	24	Signal /B from axis 2
6	Home from axis 2	25	Home from axis 2
7	Signal A from axis 3	26	Signal /A from axis 3
8	Signal B from axis 3	27	Signal /B from axis 3
9	Home from axis 3	28	Home from axis 3
10	Signal A from axis 4	29	Signal /A from axis 4
11	Signal B from axis 4	30	Signal /B from axis 4
12	Home from axis 4	31	Home from axis 4
13	HTL 1/0 threshold	32	TTL 1/0 threshold
14	Probe (Pulse)	33	Probe (GND)
15	Probe led anode	34	Probe led cathode
16	+5V	35	GND (0V)
17	+5V	36	GND (0V)
18	+12V	37	Chassis (ground)
19	+12V		

Figure 4, Feedback Connector

2.6.2 Digital input and output connector

Female SUBD type connector with 15 pins, with the following signal allocations:



Pin	Signal	Pin	Signal
1	Output 1	9	Output voltage 1
2	Output 2	10	Output voltage 2
3	Output 3	11	Output voltage 3
4	Output 4	12	Output voltage 4
5	Input 1	13	GND inputs 24V
6	Input 2	14	Not connected
7	Input 3	15	Not connected
8	Input 4		

Figure 5, Digital input and output connector

3. CONNECTION OF FEEDBACK DEVICES

The cables to be used must be shielded, with the mesh connected to the protection chassis at both ends (connectors). If they are to undergo sharp movements, or be pulled or exposed to dirt or aggressive liquids they should be surrounded by a protective sheath. If possible, use cables and extensions supplied by the manufacturer.

3.1 Connection to Fagor feedback devices

MT and CT type scales

- Non-differential TTL signals. Supply with +5V, connect the signals to their equivalents and connect the pins of the differential signals, on the connector of the DRO board, to its pin 32. Io every 50 mm (Type 0).

MX, CX and FT type scales

- Differential TTL signals. Supply with +5V, connect the signals to their equivalents. Io every 50 mm (Type 0).

MP, CP and FP type scales

- Differential 1Vpp signals. Supply with +5V, connect the signals to their equivalents. Io every 50 mm (Type 0).

MO?, CO? Type scales

- Coded Io, fixed every 20 mm (Type 1)

FO? Type scales:

- Coded Io, fixed every 100 mm (Type 3)

3.2 Connection to other feedback devices

Bear in mind the connection recommendations of the feedback device manufacturer.

In the event of the feedback device not supplying differential feedback signals, the corresponding pins on the DRO board connector must be connected to pin 13 or 32, depending on the signal level.

4. CONNECTING THE TOUCH PROBE

The Fagor DRO PC board is provided with a relay contact input for 5V probes. Depending on the type of connection applied one has to indicate, in the initialization structure of the DRO board, if the probe operates with signal logic high or low.

4.1 Connection of the contact-through-the-part probe

Simple connection, PAM-10 type Renishaw probe:

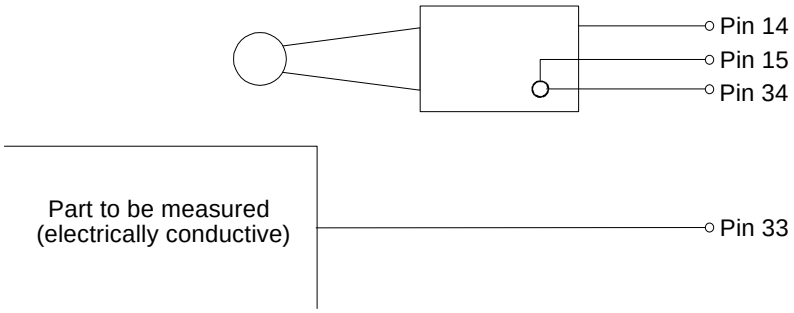


Figure 6, Simple connection of the probe

With this type of connection operation is with low logic.

4.2 Connection of an internal contact probe

Probe with normally open contact output:

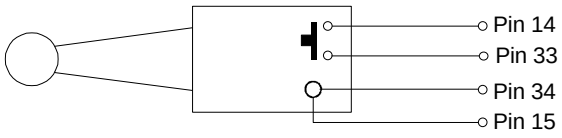


Figure 7, Connection with a normally open contact

With this type of connection, operation is with low logic.

Probe with a ‘normally closed’ contact output:

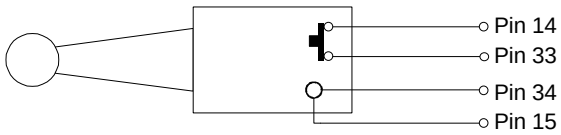


Figure 8, Connection with a normally closed contact

With this type of connection, operation is with logic high.

5. CONNECTION OF DIGITAL INPUTS AND OUTPUTS

Use the SUBD 15-pin connector supplied for connecting the inputs and outputs to the electrical cabinet. Use a shielded cable, with wires of at least a 0.14 mm^2 section, depending on the current to run through them.

The input diagram is as follows:

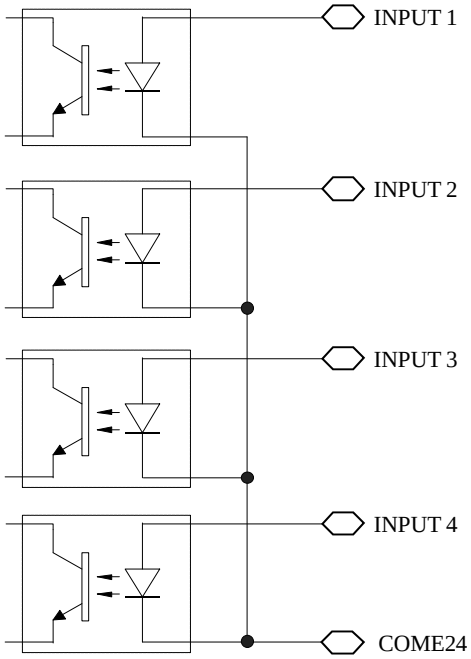


Figure 9, Diagram of digital inputs

The diagram of each output is :

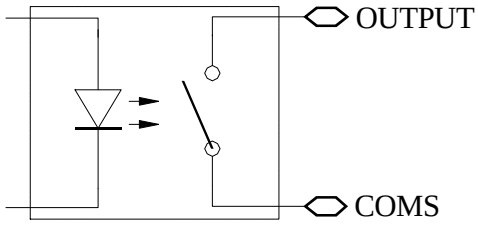


Figure 10, Diagram of digital outputs

For each output, the following anti-interference circuit should be externally connected:

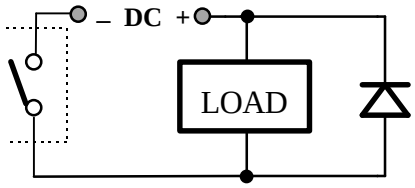
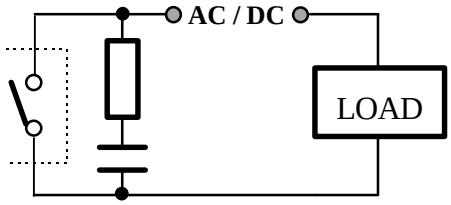
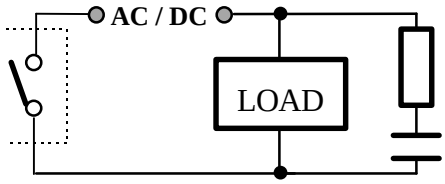
DC LOAD		**Diode** V_{BR} = 2 to 4 times VDC $I_F = I_L$ maximum (1N4000 series)
AC/DC LOAD	 	**Resistor** $R \approx R_{LOAD}$ 1W **Capacitor** 0.1…1 µF 250V

Figure 11, Connection of Digital outputs

6. SUPPLIED FUNCTIONS

6.1 Data structures used

The following data structures are defined in file '**fdrodos.h**' for applications in DOS and '**fdrow16.h**' for Windows 3.x applications.

- Setting data for each axis :

```
typedef struct
{
        double          Resolution;
        BOOLEAN         CountDirection;
        unsigned int    TTLWaveMultiply;
        unsigned int    SignalType;
        unsigned int    SineWaveMultiply;
        double          MachineErrorComp;
        unsigned int    I0Type;
} AxisConfig_t;
```

where *Resolution* contains the axis resolution. The value can be in any units, millimeters or inches, or even values for rotary axes. The count function is only used for returning a significant value to the probe treatment functions.

CountDirection can be 0 or 1. A zero value indicates assuming counting with the default sign. A value of one indicates changing the count sign.

TTLWaveMultiply indicates the multiplication factor that should be applied to the counting. A zero value means a x4 multiplication (default value), one x2, two x1, and so on.

SignalType indicates the type of signal supplied by the feedback device. For TTL and HTL signals one has to put a value of zero, and a value of one for 1Vpp signals.

SineWaveMultiply indicates the value of the subdivision to be applied in the case of 1Vpp sine wave signals. Values between 1 and 255 are accepted. It also has to contain the externally applied multiplication, when the signal is TTL and the type of home marker pulse (I_0) is coded. For TTL signals with no coded I_0 a value of one should be set.

MachineErrorComp indicates the value, in the same units as the resolution, for the table sag compensation. This value is used as follows:

*counting = counting * (1 + MachineErrorComp)*

for example, if one wishes to compensate for a table sag error of 10 µm per meter a value of '0.01 / 1000', that is, *0.00001* has to be entered. If the compensation required is one ten thousandth of an inch per foot, and the axis has resolution in inches, '0.0001 / 12', that is *0.000008333* has to be entered.

I0Type indicates the type of reference mark (marker) of the feedback device. A value of zero indicates that there are no I_0s or that these are normal, and applies both to linear and rotary feedback devices. The following values are applied when the feedback device is linear with coded I_0s :

Value	Meaning
1	Distance of 20 mm and increasing
2	Distance of 20 mm and decreasing
3	Distance of 100 mm and increasing
4	Distance of 100 mm and decreasing

- Counting data and input values. It is also where the value for the outputs is placed:

```
typedef struct
{
        unsigned int    InputsValue;
        unsigned int    OutputsValue;
        double          AxisDisplayValue[4];
} BoardValues_t;
```

InputsValue contains the value of the digital inputs, as read in the last call to the function *ReadAll Counters*.

OutputsValue contains the value of the digital outputs, as written in the last call to the function *ReadAllCounters*.

AxisDisplayValue contains the value of the count for the four axes, as read in the last call to function *ReadAll-Counters*. This value is not the number of pulses, but already has the resolution applied, as well as the direction, multiplication and table sag error compensation for the axis and indicated in the data structure *AxisConfig_t*.

- Board configuration data (only applicable under Windows) :

```
typedef struct
{
        unsigned int PtrBoardBase;
        unsigned int I0IrqNumber;
        unsigned int ISA_Tic;
        unsigned int NumberOfAxes;
        unsigned char dummy[12];
} BoardConfig_t;
```

PtrBoardBase contains the board access address.

I0IrqNumber contains the interrupt number used for seeking the references of the feedback device (I_0s) and probing points.

ISA_Tic not used, should be zero.

NumberOfAxes contains the number of axes to be handled.

dummy not used, can have any value.

6.2 Functions supplied

IniBoard

#include “fdrodos.h”
BOOL __far __pascal IniBoard(unsigned int axesnum, AxisConfig_t __far axiscfg[]);

or

#include “fdrow16.h”
BOOL __far __pascal IniBoard(BoardConfig_t *boardx, AxisConfig_t __far axiscfg[]);

Initializes the board and the internal counters according to the data for each axis. This function has to be executed before any other and only has to be called up once in each session. It returns ‘*FALSE*’ if it was not possible to initialize the board or ‘*TRUE*’ if the operation was correct.

An example :

We have two axes: the first is a 20µm pitch feedback device with TTL signal, the second is a 100µm pitch feedback device with 1Vpp signal in which we require a resolution of 5 µm. The values to be entered are :

```
axiscfg[0].Resolution =
axiscfg[1].Resolution = 0.005;        /* both 5 µm */

axiscfg[0].CountDirection =
axiscfg[1].CountDirection = 0;

axiscfg[0].TTLWaveMultiply =
axiscfg[1].TTLWaveMultiply = 0;   /* x4 */

axiscfg[0].SineWaveMultiply = 1;   /* TTL signal*/
axiscfg[1].SineWaveMultiply = 5;   /* x5 */

axiscfg[0].MachineErrorComp =
axiscfg[1].MachineErrorComp = 0;

axiscfg[0].I0Type = 0;
```

```
axiscfg[1].I0Type = 3;

if(IniBoard(2, axiscfg) == FALSE)
                        initialization_error();
```

ReadBoardId

#include “fdrodos.h” or #include “fdrow16.h”
BOOL __far __pascal ReadBoardId(char __far *id_ptr);

Each DRO board has a unique identification code. This function is supplied to read said code, which can be used as protection for the application or simply to check the existence of the DRO board. The stream ‘id_ptr’ should have a minimum length of 17 characters, as the function leaves 16 plus the end of stream character. The function sends back ‘*TRUE*’ if the reading was correct or ‘*FALSE*’ if it could not be read.

For example :

```
char idplaca[17];

if(ReadBoardId(idplaca) == FALSE)
                              identificator_error();
```

ReadAllCounters

#include “fdrodos.h” or #include “fdrow16.h”
void __far __pascal ReadAllCounters(BoardValues_t __far *ptvalue);

This function reads the counters of all the active axes, carries out the resolution, direction and table sag compensation calculations to send back the position of the axis in the units configured. It also reads the value of the digital inputs and updates that of the outputs. This function should be called up every time one wishes to find out the real position of the axes. The most usual process is to call this up from a periodic interruption. Furthermore, if the probe mode is active, it checks if this is in contact with the part or not, by controlling the probe led coming on and going off.

LookForI0

#include “fdrodos.h” or #include “fdrow16.h”
BOOL __far __pascal LookForI0(unsigned int axis);

After switching on the PC, the value of the counters is unpredictable. If you wish to recover the value of the axis position, a feedback device reference (home) search has to be made. This function activates the interrupt mode to detect the reference mark of the selected axis.

IfI0Found

#include “fdrodos.h” or #include “fdrow16.h”
BOOL __far __pascal IfI0Found(void);

This function is used to find out if the reference mark in the selected axis has been found by the previous function. It sends back ‘*TRUE*’ if the reference mark has already been overshot or ‘*FALSE*’ if this has not yet been reached.

EndLookForI0

#include “fdrodos.h” or #include “fdrow16.h”
void __far __pascal EndLookForI0(void);

Whether the feedback device reference point has been found or the search is to be canceled, this function has to be called up. This deactivates the interrupt mode and restores the normal counting mode.

StartProbeMode

#include “fdrodos.h” or #include “fdrow16.h”
BOOL __far __pascal StartProbeMode(int mode);

To carry out probing functions with a edge probe one should first activate this function. It carries out the preparation of the points capture zone and the entry in interrupt mode for probing. Parameter *mode* has the following meaning.

Value	Meaning
0	Probe activation logic low (down flank).
1	Probe activation logic high (up flank).

GetProbePoint

#include “fdrodos.h” or #include “fdrow16.h”
int __far __pascal GetProbePoint(BoardValues_t __far *ptvalue);

Reads the position of the axes, as well as the status of inputs, at the time a probe pulse is activated. Uses the same type of data as the function *ReadAllCounters*, although the value of the outputs is not used.

It also checks if the probe is in contact with the part or not, by controlling when its led comes on or goes off. It sends back :

Value	Meaning
0	There are no probing points, the position is not significant.
1	This is the last point.
2	There are further probing points not read.
-1	There was an overflow of the storage zone or some other error during probing - the position is not significant. .

LedProbe

#include “fdrodos.h” or #include “fdrow16.h”
int __far __pascal LedProbe(int mode)

Checks or changes the status of the probe led according to the value of parameter *mode*, sends back the current status of the led. The values that can be assumed by the parameter and the return are::

Value	Meaning
0	Only the parameter, to find out what the status of the led is.
1	Led on.
-1	Led off.

EndProbeMode

#include “fdrodos.h” or #include “fdrow16.h”
void __far __pascal EndProbeMode(void);

Exits the probing mode, although there may still be points to be read.

7. BUILDING AN APPLICATION

7.1 Programming in C language

The library *fdrodos.lib* is generated with Microsoft C++ version 1.52 compiler, for working in MS-DOS with a 286 or later microprocessor.

The functions are defined as *__pascal* for maintaining compatibility with the Windows version.

The library *fdrow16.dll* is generated for working in Windows versions 3.x. Library *fdrow16.lib* should be used in order for error : *unresolved externals* not to come up when linking the application.

7.2 Program example

DEMO.EXE

This program, including the source code in C language, carries out all the functions for accessing the DRO board from DOS.

In the center of the screen, it displays the count for the four axes.

On the lower line, it displays the status of the inputs, the present value of the outputs, the count units (inches or millimeters) and the board identification code.

Meaning of the keys in execution:

i Toggles the count display, between millimeters and inches.

axis When the letter for the name of the axis is pressed we go into the home reference search for the feedback device of the axis selected. To exit without carrying out the search press key **c**.

r Resets the count displayed for all the axes.

p Changes to working with probe mode. To exit this mode press key **c**.

1
2
3
4 Changes the value of the relevant output. If this is zero it changes it to one and vice versa.
9 Makes all the outputs active
0 Makes all the outputs inactive

<esc> To exit the program.

APPENDIX A, PC input/output address map

Address (Hex)	Function
000 - 1FF	Basic System
200	Reserved
201	Game port
202-277	Free
278-27F	LPT2:
280-2F7	Free
2F8-2FF	COM2:
300-377	Free
378-37F	LPT1:
380-3AF	Free
3B0-3BF	Monochrome display adapter
3C0-3CF	Reserved
3D0-3CF	Graphics/ Color adapter
3E0-3EF	Reserved
3F0-3F7	Diskette drive
3F8-3FF	COM1:

APPENDIX B, System interrupts

Interruption	Usually
3	COM2:
5	LPT2:
7	LPT1:

APPENDIX C, Internal access addresses

Offset	R / W[2]	Description
0x00	R	Counter for axis 1 (4 readings = 32 bits) format bytes = HI MH ML LO
0x01	R	Status of axis 1 counter. Bit 5 = count direction (0 = positive), the rest are not significant.
0x02	R	Counter for axis 2 (4 readings = 32 bits)
0x03	R	Status of axis 2 counter
0x04	R	Counter for axis 3 (4 readings = 32 bits)
0x05	R	Status of counter for axis 3
0x06	R	Counter for axis 4 (4 readings = 32 bits)
0x07	R	Status of counter for axis 4
0x08	R	A/D converter, value of signal A
0x09	R	A/D converter, value of signal B
0x0A	R	Probe level (0 / 1)
0x0B	R	Identification code (8 bytes hex, series reading)
0x0C	R	Excess in conversion signal A (0 / 1)
0x0D	R	Excess en conversion signal B (0 / 1)
0x10	R	Inputs of expansion board 1 (4 low bits)
0x18	R	Inputs of expansion board 2 (4 low bits)
0x00	W	Clearing of internal counters (Needs later latching to make zeroing effective). Type a '1' on the bit for the axis (43214321b).
0x01	W	Software latching of internal counters. Type '0xF0' followed by '0x0F' before reading the counters of the four axes successively or only the active bits of the axes to be read.
0x02	W	Selection of internal count multiplication. Always '0xFF'. The counter does an x4,and if less is required, do it via software.
0x03	W	Selection of the internal latch masks. Always '0xFF'
0x04	W	Selection of internal configuration. Always

(2) **R** means reading and **W** writing.

Offset	R / W[2]	Description
		'0x0F'.
0x08	W	Selection of signal to be converted (A/D) 0 to 3 = axis 1 to 4 4 = I_0s axes 1 and 3[3] 5 = I_0s axes 2 and 4[3] 6 & 7 = do not convert
0x09	W	Selection of the interruption source 0 = none 1 to 4 = I_0 axis 1 to 4 5 = probe 6 = external signal 7 = none
0x0A	W	Probe led on or off (1 / 0)
0x0B	W	Writing to be able to read the identification code
0x0C	W	Selection of the flank or level for interrupt (0 / 1)
0x0D	W	Resetting of the interruption
0x0E	W	Generation of counter latching with no interruption
0x0F	W	Resetting counter latching with no interruption
0x10	W	Outputs of the expansion board 1 (4 low bits)
0x18	W	Outputs of the expansion board 2 (4 low bits)

(3) Analog values will read the first I_0 as signal A and the second as signal B